

IN THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously Presented) A computer-implemented method for generating a configurator comprising:
creating a customizable product class, the customizable product class including a set of one or more attributes to define the customizable product class;
adding a component product class to the customizable product class, the component product class is a subclass of the customizable product class; and
mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator,
wherein the customizable product class is to represent a consumer product and the component product class is to represent one or more components of the consumer product.
2. (Original) The method of claim 1 wherein the component product class includes component product subclasses.
3. (Original) The method of claim 1 wherein the component product class inherits the attributes of the customizable product class.
4. (Original) The method of claim 1 further comprising:
adding one or more component product classes to a port; and
adding the port to the customizable product class, the port to allow the configurator to classify a group of component products.

5. (Original) The method of claim 4 wherein the port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.
6. (Original) The method of claim 5 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.
7. (Original) The method of claim 5 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.
8. (Original) The method of claim 1 wherein the mapping to include building the customizable UI from a set of themes, groups, and controls.
9. (Original) The method of claim 8 wherein the themes are tabs and wizards.
10. (Original) The method of claim 8 wherein each theme in the set of themes, groups, and controls includes at least one of the set of background colors, fonts, and multi-linguals.
11. (Original) The method of claim 8 wherein the group includes one or more of the controls.
12. (Original) The method of claim 8 wherein the control includes one or more of a drop down box, a radio button, and a list box.
13. (Original) The method of claim 1 wherein the customizable UI is used to generate a user interface for a component product class.
14. (Original) The method of claim 1 wherein the customizable UI is a subclass of the customizable product.

15. (Original) The method of claim 1 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.
16. (Original) The method of claim 1 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.
17. (Original) The method of claim 1 wherein the component product class, customizable class rules, and UI class are object oriented classes.
18. (Original) The method of claim 1 wherein the customizable product has an object oriented structure.
19. (Original) The method of claim 1 wherein the customizable product includes versioning.
20. (Original) The method of claim 1 wherein the configurator is stored in a data store.
21. (Previously Presented) A machine-readable medium that provides instructions, which when executed by a set of one or more processors, cause the set of processors to perform operations for generating a configurator comprising:
creating a customizable product class, the customizable product including a set of one or more attributes to define the customizable product;
adding a component product class to the customizable product class, the component product class is a subclass of the customizable product; and
mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator,
wherein the customizable product class is to represent a consumer product and the component product class is to represent one or more components of the consumer product.

22. (Original) The machine-readable medium of claim 21 wherein the component product class includes component product subclasses.
23. (Original) The machine-readable medium of claim 21 wherein the component product class inherits the attributes of the customizable product class.
24. (Original) The machine-readable medium of claim 21 further comprising:
adding one or more component product classes to a port; and
adding the port to the customizable product class, the port to allow the configurator to
classify a group of component products.
25. (Original) The machine-readable medium of claim 24 wherein the port includes a
cardinality attribute, the cardinality attribute to constrain the number of component
products to be added by the configurator.
26. (Original) The machine-readable medium of claim 25 wherein the cardinality
attribute includes a minimum cardinality and a maximum cardinality, the minimum
cardinality to constrain the minimum number of component products to be added by
the configurator, the maximum cardinality to constrain the maximum number of
component products to be added by the configurator.
27. (Original) The machine-readable medium of claim 25 wherein the cardinality
attribute includes a default cardinality, the default cardinality defines a quantity of the
component product class added by the configurator.
28. (Original) The machine-readable medium of claim 21 wherein the mapping to include
building the customizable UI from a set of themes, groups, and controls.
29. (Original) The machine-readable medium of claim 28 wherein the themes includes
tabs and wizards.
30. (Original) The machine-readable medium of claim 28 wherein the theme includes
background color, fonts, and multi-lingual.

31. (Original) The machine-readable medium of claim 28 wherein the group includes one or more of the controls.
32. (Original) The machine-readable medium of claim 28 wherein the control includes one or more of a drop down box, a radio button, and a list box.
33. (Original) The machine-readable medium of claim 21 wherein the customizable UI is used to generate a user interface for a component product class.
34. (Original) The machine-readable medium of claim 21 wherein the customizable UI is a subclass of the customizable product.
35. (Original) The machine-readable medium of claim 21 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.
36. (Original) The machine-readable medium of claim 21 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.
37. (Original) The machine-readable medium of claim 21 wherein the component product class, customizable class rules, and UI class are object oriented classes.
38. (Original) The machine-readable medium of claim 21 wherein the customizable product has an object oriented structure.
39. (Original) The machine-readable medium of claim 21 wherein the customizable product includes versioning.
40. (Original) The machine-readable medium of claim 21 wherein the configurator is stored in a data store.
41. (Previously Presented) An object oriented configurator comprising:
a customizable product class;

a component product, the component product is a subclass of the customizable product, the component product inherits a set of one or more attributes from the customizable product class; and

a customizable UI, the customizable UI is mapped to the customizable product providing a view of the component product,

wherein the customizable product class is to represent a consumer product and the component product class is to represent one or more components of the consumer product.

42. (Original) The object oriented configurator in claim 41 further comprising:
a port, the port comprising a set of one or more of the component products.
43. (Original) The object oriented configurator in claim 42 wherein the port includes a cardinality, the cardinality to constrain the number of component products to add to the customizable product class.
44. (Original) The object oriented configurator in claim 43 wherein the cardinality includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.
45. (Original) The object oriented configurator in claim 43 wherein the cardinality includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.
46. (Original) The object oriented configurator in claim 41 the customizable class rule, and customizable UI are subclasses of the customizable product.

47. (Original) The object oriented configurator in claim 41 wherein the component product includes a static attribute, the static attribute is not inherited from a parent class.
48. (Original) The object oriented configurator in claim 41 wherein the attribute is of type string, number, date, and Boolean.
49. (Original) The object oriented configurator in claim 41 further comprising a second customizable product.
50. (Original) The object oriented configurator in claim 49 wherein the component product includes one or more of a second customizable product.
51. (Original) The object oriented configurator in claim 41 wherein the component product includes an expression to restrict the component product from becoming a subclass of the customizable product class.
52. (Original) The object oriented configurator in claim 41 further comprising:
a script, the script to communicate with another application.
53. (Original) The object oriented configurator in claim 41 wherein the customizable UI includes a theme, group, and control.
54. (Original) The object oriented configurator in claim 41 wherein the theme includes a tab, wizard, font, and color.
55. (Original) The object oriented configurator in claim 41 wherein the control includes one or more of a drop down box, a radio button, and a list box.
56. (Original) The object oriented configurator in claim 41 wherein the customizable UI map comprises HTML, JAVA applets, and ActiveX components.

57. (Original) The object oriented configurator in claim 41 wherein each component product class has an unique identifier, the unique identifier is used to locate an associative component product.
58. (Original) The object oriented configurator in claim 41 further comprising link items.
59. (Currently amended) An apparatus ~~composed of logic blocks to customize a product~~ comprising:
a ~~first logic block to create~~ means for creating a customizable product class, the customizable product class including a set of one or more attributes to define the customizable product class;
a ~~second logic block to add~~ means for adding a component product class to the customizable product class, the component product class is a subclass of the customizable product class; and
a ~~third logic block to map~~ means for mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator,
wherein the customizable product class is to represent a consumer product and the component product class is to represent one or more components of the consumer product.
60. (Previously Presented) The apparatus of claim 59 wherein the component product class includes component product subclasses.
61. (Previously Presented) The apparatus of claim 59 wherein the component product class inherits the attributes of the customizable product class.
62. (Currently amended) The apparatus of claim 59 further comprising:

~~a fourth logic block to add~~ means for adding one or more component product classes to a port; and

~~a fifth logic block to add~~ means for adding the port to the customizable product class, the port to allow the configurator to classify a group of component products.

63. (Previously Presented) The apparatus of claim 62 wherein the port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.
64. (Previously Presented) The apparatus of claim 63 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.
65. (Previously Presented) The apparatus of claim 63 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.
66. (Currently amended) The apparatus of claim 59 wherein the means for mapping ~~third logic block to map to~~ includes means for building the customizable UI from a set of themes, groups, and controls.
67. (Previously Presented) The apparatus of claim 66 wherein the themes are tabs and wizards.
68. (Previously Presented) The apparatus of claim 66 wherein each theme in the set of themes, groups, and controls includes at least one of the set of background colors, fonts, and multi-linguals.
69. (Previously Presented) The apparatus of claim 66 wherein the group includes one or more of the controls.

70. (Previously Presented) The apparatus of claim 66 wherein the control includes one or more of a drop down box, a radio button, and a list box.
71. (Previously Presented) The apparatus of claim 59 wherein the customizable UI is used to generate a user interface for a component product class.
72. (Previously Presented) The apparatus of claim 59 wherein the customizable UI is a subclass of the customizable product.
73. (Previously Presented) The apparatus of claim 59 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.
74. (Previously Presented) The apparatus of claim 59 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.
75. (Previously Presented) The apparatus of claim 59 wherein the component product class, customizable class rules, and UI class are object oriented classes.
76. (Previously Presented) The apparatus of claim 59 wherein the customizable product has an object oriented structure.
77. (Previously Presented) The apparatus of claim 59 wherein the customizable product includes versioning.
78. (Previously Presented) The apparatus of claim 59 wherein the configurator is stored in a data store.